Real Time Tempo Canons with Antescofo

Christopher Trapani, José Echeveste

Real Time Tempo Canons with Antescofo

Christopher Trapani
Columbia University, New York
cmt2150@columbia.edu

José Echeveste
IRCAM UMR STMS 9912 - CNRS
Sorbonne Universités UPMC - INRIA Rocquencourt
echeveste@ircam.fr

ABSTRACT

With recent advances in score-following capabilities, it has become possible to envision new timing strategies, to realize previously impractical methods of coordination between a live performer and electronics. Our work centers on the challenge of synchronizing at the end of a musical phrase, where events and processes are timed not from an initial trigger, but occur according to a relative distance towards a predicted future attack.

The key software component is *Antescofo*, a score-following tool which relies on a strong coupling, through a dynamic programming language, between a machine listening module and a reactive engine. This language allows hierarchical, concurrent, and heterogeneous computer processes to be organized over time, and for external events to be anticipated, with a runtime system that triggers electronic actions in response to variations in performance.

Various programming strategies were implemented, honed and tested with a live performer. The musical material of these sketches focused on the idea of flexible canons, on interactions between a live instrument and a second voice generated from a buffered real-time recording. The different canonic strategies make use of *Antescofo*'s live tempo calculations to create dynamic tempo shifts, to force voices to converge, and to control precise canon effects.

1. INTRODUCTION

Electroacoustic performance has long relied on the method of front-end synchronization, allowing live players to interact with electronics by triggering events in real time, with processes timed to unfold from these interspersed cues. With a reliable score-following tool coupled with a domain specific language, different strategies of synchronization have become feasible, notably the possibility of aligning at the end of a musical phrase.

Using the tempo canon principles of Conlon Nancarrow ([1],[2]) as a model, voices at related but independent speeds catch up at a predetermined convergence point (Fig 1). Unlike Nancarrow's rigid grids however, a degree of rhythmic flexibility and vitality becomes possible with the dynamic score-following tool *Antescofo*, whose programming structure allows high-level macro controls that adapt to real-time tracking data, adapting the playback speed of the canons accordingly to accompany the inevitable fluctuations of a live performance. Electronic events are timed no longer from a given starting point, but in relation to an approaching moment whose distance is continually predicted and recalculated.

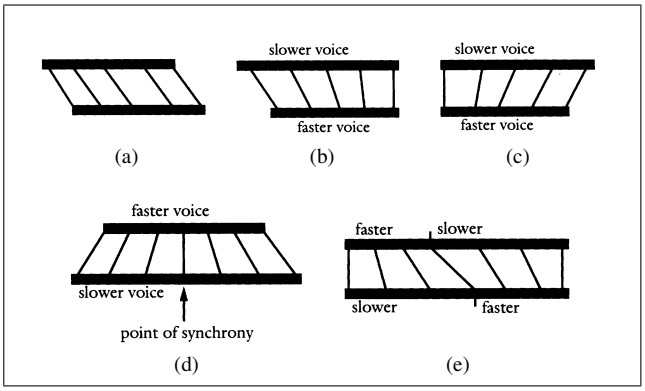

Figure 1: (1a) conventional canon. (1b) converging canon. (1c) diverging canon. (1d) converging diverging canon. (1e) diverging converging canon.

2. ANTESCOFO

2.1 Synchronizing electronic actions with a musician performance

Research around the *Antescofo* system focuses on how to achieve a high-level musical interaction between live musicians and a computer, specifically in the context of mixed music, where the temporal development of musical processes depends on active listening and complex synchronization strategies [3].

We have proposed in [4] a novel architecture (Fig 2) that relies on a strong connection between artificial machine listening and a domain-specific real-time programming language for compositional and performative purposes. To instruct the artificial listening engine, the user creates an *augmented score* whose language integrates both programming tools and musical terms, allowing a unique and flexible temporal organization. The augmented score outlines both the instrumental and the electronic parts and the instructions for their real-time coordination during performance, the specific events that should be recognized in real

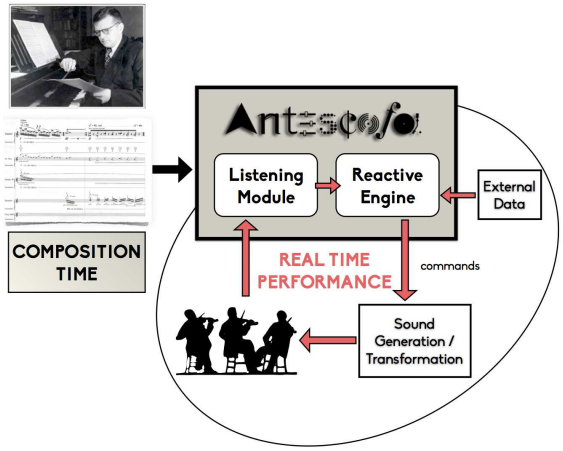

Figure 2: General architecture of Antescofo. The score is an input of both the listening machine and the reactive engine. The events recognized by the listening machine are signaled to the reactive engine which schedule the actions at the right time.

time, and the actions corresponding to these triggers. During a performance, the language runtime evaluates the augmented score and controls processes synchronously with the musical environment, with the aid of artificial machine listening. This approach has been implemented in the *Antescofo* system and validated in many live performances.

2.2 Some elements of the language

An *Antescofo* score contains both an instrumental part and the accompaniment actions. As explained above, details of both notated music and electronics are combined in a single augmented score. During live performance, *Antescofo* reacts to data from the listening machine and from the external environment. This reactive system dynamically takes account of tempo fluctuations and the values of external variables, in order to synchronize accompaniment actions to the real-time interpretation. The possibility of *timing events and the actions relative to the tempo*, as in a classical score, is one of the main strengths of *Antescofo*. Within the augmented score language, the composer can decide to associate actions to certain events with delays (either in absolute time or relative to the tempo), to group actions together, to define timing behaviors, to structure groups hierarchically, and to allow these groups to act in parallel.

An augmented score is a sequence of instrumental events and actions. The syntax for the instrumental part allows the description (pitches + duration) of events such as notes, chords, trills, glissandi and improvisation boxes [4]. Actions are divided into *atomic actions* which perform an elementary computation and *compound actions*. Compound actions group multiple actions; these are triggered more or less directly by an event.

```
NOTE C4 1.0
  1/2 receive_action $v1 $v2
CHORD (D4 F#4) 1/2
      receive_action $v3 $v4
```

This excerpt from an *Antescofo* score shows two musical events with two electronic actions. The first corresponds to a single note (middle C) with a duration of one beat. An electronic action is triggered by *Antescofo* one eighth-note after the C is detected, where the message "`$v1 $v2`" is sent to the object `receive_action`.

3. REALIZATION

Sketches written by Christopher Trapani and played by the clarinetist Jérôme Comte from the *Ensemble intercontemporain* formed the basis of a six-month study, culminating with a 3-minute final sketch viewable here [5]. In this section we will detail the various challenges addressed during this collaboration.

3.1 Set-up

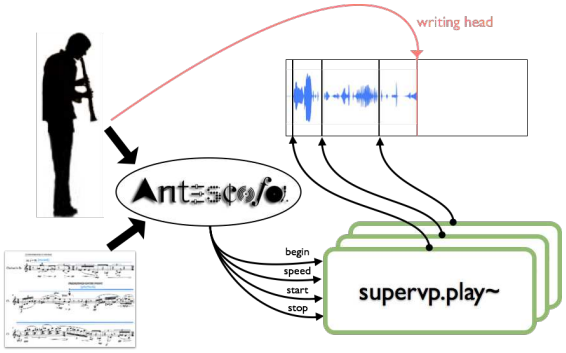

Figure 3: *Antescofo* follows the clarinetist's performance which is recorded in parallel into a buffer. Phase Vocoder objects read this buffer at specific positions and speeds controlled by *Antescofo* as outlined in the score.

These sketches were realized in the Max/MSP environment. The live performer is recorded into an audio buffer, while a Phase Vocoder object (*SuperVP* [6], [7]) reads this buffer at specific positions and speeds. The *Antescofo* program calculates the correct position and speed values according to incoming real-time data, communicating with the Phase Vocoder.

The challenge of creating a dynamic canon lies in building, estimating, and updating in real time the relation between the temporality of the score (in beats) and absolute time. All parameters that control the unfolding of the canon depend on this relationship. The *Antescofo* language allows the composer to manage this kind of control in a precise fashion.

3.2 Description of Real-Time Canons

A canon can be defined as the superimposition of a phrase upon a version of itself. In our work, the live musician provides the first voice of the canon, whereas the electronics provide a manipulated second voice in reply. A two-voice canon can thus be characterized as a symbiotic phrase with two components:

- the canonic reply played back from a buffer (the line segment [start,end] in the figures 4),

- the phrase performed live by the musician onto which the canonic reply is superimposed (the line segment [start',sync] in the figures 4).

Keeping in mind that the playback position can never overtake that of the live recording for obvious causal reasons, different types of real-time canons can be defined (Fig 4).

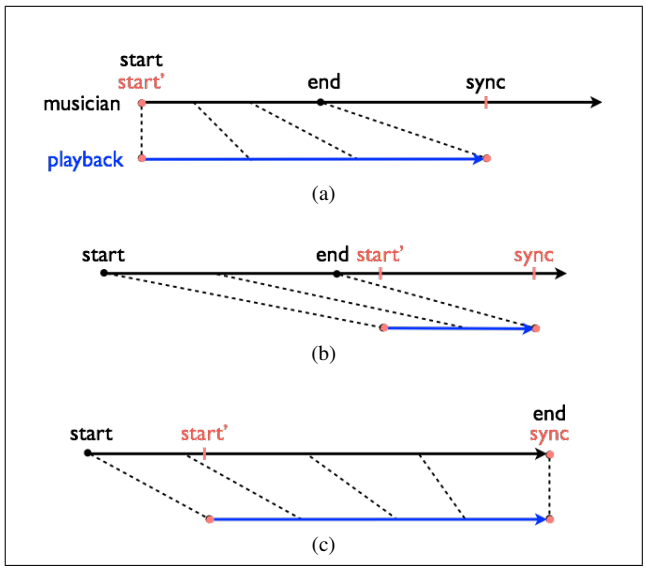

Figure 4: Different types of real-time canons

The temporal data pertaining to each canon is specified in the score. These specifications are virtual and relative to the tempo so that during performance, these values can be translated into a real and absolute scale.

The canon (4a) starts at the same time as the musician but unfolds at a slower speed. The canon (4b) starts after the musician has played the end of a phrase. This corresponds to a superposition of two independent voices. The canon (4c) begins after the live voice at a slower tempo that will gradually increase until catching up the musician at a predefined convergence point.

3.3 Canon speed calculation

In order to determine the precise speed for each canon in the score, the *Antescofo* program uses real-time tempo calculations as well as temporal estimations of upcoming events. The tempo of the musician as estimated by the listening machine of *Antescofo* is the key to calculating these estimates of future events. Thus, the speed values are re-estimated throughout the entire duration of the canon as soon as new information concerning the position or the tempo of the musician arrives.

Figure 5 represents the different relationships which can be found in the three types of canons mentioned above.

Let t the current position of the musician in the absolute scale; the computation of the speed follow this equation:

$$canonSpeed = \frac{end - t'}{sync - t} \quad (1)$$

where $sync - t$ is the estimation of the time before the synchronization point and $end - t'$ is the estimation of the

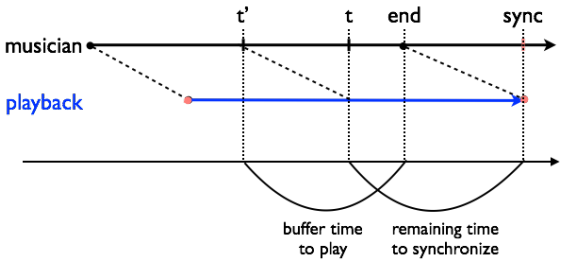

Figure 5: The diagram represents the different values to be considered for the speed calculation. At the current time t, the dates end (the canon end) and $sync$ (the final synchronization point) are estimated according to the tempo value. The t' value corresponds to the time at which the live musician played the current position of the buffer.

time that remains to be played by the canon until the synchronization point.

This is the main piece of code in the score that computes the speed of the canon each time the tempo is updated:

```
whenever($RT_TEMPO)
{
 $t':=$t'+((@date($t')-$NOW)*$speed)

 $end:=$NOW+($end_pos-$RNOW)*60/$RT_TEMPO
 $sync:=$NOW+($sync_pos-$RNOW)*60/$RT_TEMPO

 $speed := ($end-$t')/($sync-$NOW)
 ph_voc speed $speed

}until ($RNOW ≥ $end_pos)
```

`$NOW`, `$RNOW`, and `$RT_TEMPO` are internal variables that represent, respectively, the current time in seconds, the current position of the musician in beats, and the current value of the tempo.

The `whenever` statement allows actions to be launched when a given condition is verified. Each time the variables corresponding to this condition are updated, the predicate is re-evaluated. Here the body of the `whenever` is executed each time the variable `$RT_TEMPO` changes. This process continues until the live musician has reached the end of the canon (`$RNOW $\geq$ $end_pos`).

`$t'` represents the audio time-chunk in the buffer already played; `@date(t')` is the date of the last update of `$t'` variable. The tempo estimation is used to predict the dates of `$end` and `$sync`. `$speed` represents the speed of the playback and is used as the control parameter of the phase vocoder.

3.4 Real-Time Canons with speed curve constraints

Traditionally, canons are defined by a constant speed ratio. Phase vocoder playback however offers the possibility of dynamic tempo flexibility following a predefined pattern. These shifts in speed can be easily stored as a function graph, with care taken to avoid increased playback speeds that would cause the playback buffer to overtake the recording. Equally applicable to prolongation and converging canons, these tempo maps provide a function trans-

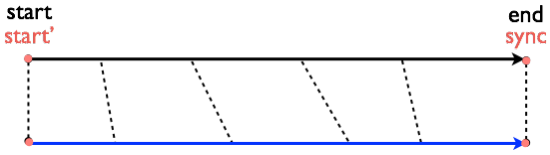

Figure 6: Real-Time Canons with speed curve constraint

fer, an intermediary multiple that imposes a second set of speed relationships onto the canonic reply.

This strategy is also used to create a specific style of canon (type e in Fig. 1) where two voices cover the same length, both beginning and ending together, but with dynamic changes of speed within this frame. Nancarrow makes frequent use of this effect, switching the speeds between a faster and a slower voice at the exact midpoint of the phrase for a flawless alignment. In our example, the tempo shifts dynamically. From an initial shared starting point, the playback voice unfolds at a slower speed than the live player's, giving the impression of a prolongation canon with a malleable tempo. Later in the canon, phrases in playback are repeated faster than their live renditions, with the tempo again continually calibrated against the approaching convergence point.

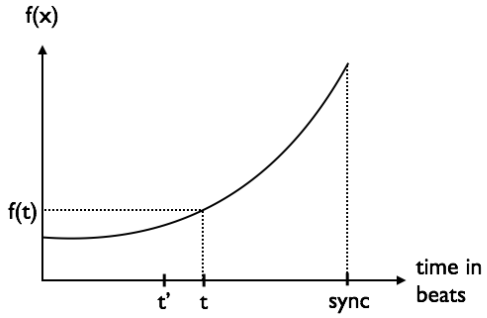

Figure 7: Example of a curve specified by the composer to constraint the speed of the canon.

The figure 7 shows a possible curve to constraint the speed of the canon. If t is the relative position of the musician in the score, the sent speed is:

$$canonSpeed = f(t) * \frac{sync - t'}{\int_t^{sync} f(x)\,\mathrm{d}x} \qquad (2)$$

where f(t) is the current value of the curve, $sync - t'$ is the time in beats that remains to be played by the canon until the synchronization point and $\int_t^{sync} f(x)\,\mathrm{d}x$ the integral of the curve from the current instant to the synchronization point.

In practice, the speed calculation is updated every 0.05 seconds.

3.5 Metadata interactions

Attack recognition data gathered in real time by *Antescofo* also serves as a tool for storing and recalling the positions of various events. As the live voice is followed, the timing in ms of each detected event is stored as a marker in the buffer. Though playback speed is continually in flux, the positions of these markers remain constant and function as signposts for various meta-processes.

As one basic example, these markers can be read as multiple entry points, so that canonic replies can begin not only at the start of the recorded buffer, but at any recognized event within the musical phrase. They may also serve as signposts for dynamic transposition, sending predetermined transposition values to the phase vocoder when the playback voice reaches the matching phrase.

Markers are also valuable for managing data outside of the signal realm. Using the onset and pitch data for each note in an Antescofo score, it is possible to keep track of the current pitch of each canonic voice, even as the playback rate changes. In our example, a process of adding and subtracting the frequencies of active pitches is established, with the results displayed in real time using the Bach interface [8]. The resulting pitches are used to pilot the transposition of concatenative synthesis in CataRT [9], producing a constantly shifting halo of samples whose harmonic content conforms to the sum and difference tones of the current pitches in the two canonic voices.

3.6 Control at the Convergence Point

The most significant challenge of creating converging canons is how to treat the arrival point, since perfect synchrony between a live voice and a buffer is impossible. One short-term solution involves a last second crossfade in playback, from the buffer to the live voice. While this allows the final attack to be reliably aligned, the drawback is a loss of precision in the passage leading up to the convergence point. The playback speed is also systematically reset to 1. (parity) at the convergence point to prevent playback from overtaking the buffer's playhead.

A watchdog mechanism is executed to prevent playback from overtaking the buffer's playhead each time the speed value of the canon is updated:

```
whenever($RT_TEMPO)
{
 abort watchdog

 ...  canon speed computation ...

 group watchdog
 {
  $delay:=($NOW-$t')/($speed-1)
  $delay  ph_voc speed 1.0
 }
}until ($RNOW ≥ $end_pos)
```

The watchdog mechanism is managed in the same body as the speed computation. The time needed for the position of the playhead (`$t'`) to catch up to the live player at the speed sent just before (`$speed`) is computed in the variable `$delay`. The message `ph_voc speed 1.0` is scheduled to be launched `$delay` seconds later. If the tempo is again updated, the `abort watchdog` command will cancel the message launch and re-schedule a new one.

The final moments of each converging canon involve increasingly perilous calculations. As the predicted arrival

point approaches, tempo calculations have more immediate consequences, and small changes in the live player's timing can create drastic alterations in playback speed. With less time left to correct course, the danger of veering ahead of the live buffer's recording point increases.

A provisional solution has been the introduction of a braking mechanism which performs a secondary prediction. If the next event anticipated by Antescofo has not yet arrived after a given percentage of the allotted window (80% by default), the speed automatically slows along an exponential curve, so that playback gradually slows until the next predicted point of recalibration arrives. A prototype braking patch was created and tested during the work period, and should be integrated into the patch after further experimentation.

4. CONCLUSION

The success of this project demonstrates *Antescofo*'s capacity for sophisticated time-related processes, handling both live performance and real-time computing through a single streamlined engine. This could open the door to novel and imaginative reinventions in the realm of rhythm, extending the existing paradigm and encouraging new experiments with ancient musical devices in a wholly contemporary context. The musical examples explored thus far have been brief and intuitive forays that only scratch the surface of the new possibilities offered by this approach.

If tempo canons can be realized in Antescofo with just a few lines of code, one can imagine a larger-scale work with several local-level tempo loops, converging and diverging in fractal-like patterns. The main barrier to composing such music today is the notation and execution of such a highly intricate level of rhythmic interplay, but a tool that generates canonic responses with real-time adaptive capabilities could lead to exciting new forays in the realm of labyrinthine rhythmic layering and detailed canonic writing.

The most important breakthrough for Nancarrow — the primary factor that paved the way for his unprecedented experimentation with multiple layers of tempo and timing — was being allowed to work out his ideas in visual and mathematical terms, directly onto the piano roll. The outcome of this project could signal a similar breakthrough in the electroacoustic realm: a new tool for intricately-timed canonic writing that could prove to be useful for many composers, enabling them to succinctly and reliably describe complex numerical relationships without recourse to notation.

Acknowledgments

This work is made possible by a grant from the French National Research Agency (ANR) INEDIT Project (ANR-12-CORD-0009).

5. REFERENCES

[1] K. Gann, *The Music of Conlon Nancarrow*. Cambridge University Press, 1995.

[2] M. E. Thomas, "Nancarrow's canons: Projections of temporal and formal structures," *Perspectives of New Music*, vol. 38, no. 106-133, Summer 2000.

[3] A. Cont, "On the creative use of score following and its impact on research," in *Sound and Music Computing*, Padova, Italy, July 2011.

[4] ——, "Antescofo: Anticipatory synchronization and control of interactive parameters in computer music," in *Proceedings of International Computer Music Conference (ICMC)*. Belfast, August 2008.

[5] (2013, April). [Online]. Available: https://www.youtube.com/watch?v=CQ1pQNy5XJk

[6] A. Roebel, "Transient detection and preservation in the phase vocoder," in *International Computer Music Conference (ICMC)*, Singapore, Singapore, Octobre 2003, pp. 247–250.

[7] A. Roebel and X. Rodet, "Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation," in *International Conference on Digital Audio Effects*, Madrid, Spain, Septembre 2005, pp. 30–35.

[8] A. Agostini and D. Ghisi, "Bach: An environment for computer-aided composition in max," in *ICMC 2012 - International Computer Music Conference*. Ljubljana, Slovenia: IRZU - the Institute for Sonic Arts Research, Sep. 2012. [Online]. Available: http://hal.inria.fr/hal-00718854

[9] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-time corpus-based concatenative synthesis with catart," in *9th Int. Conference on Digital Audio Effects (DAFx-06)*, Montreal, Canada, September 2006.